REJ10J0767-0300Z

Everywhere you imagine. RENESAS

E8 Emulator

Additional Document for User's Manual
E8 R0E000080KCE00EP1

Renesas Microcomputer Development Environment System
M16C Family / R8C/Tiny Series
Notes on Connecting the R8C/14, R8C/15, R8C/16, and R8C/17

User's Manual

Rev.3.00
Nov. 1, 2004

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/ or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

Contents

Section 1 Connecting the Emulator with the User System

Before connecting an E8 emulator (hereafter referred to as emulator) with the user system, a connector must be installed in the user system so that a user system interface cable can be connected. When designing the user system, refer to Figure 2.1, Pin Assignments of the E8 Connector, and Figure 3.1, Example of E8 connection, shown in this manual.
Before designing the user system, be sure to read the E8 emulator user's manual and the hardware manual for related MCUs.

Table 1.1 shows the recommended connector for the emulator.

Table 1.1 Recommended Connector

Type Number	Manufacturer	Specifications
2514-6002	3M Limited	14-pin straight type

Connect pins 2, 4, 6, 10, 12, and 14 of the user system connector to GND firmly on the PCB. These pins are used as electrical GND and to monitor the connection of the user system connector. Note the pin assignments of the user system connector.

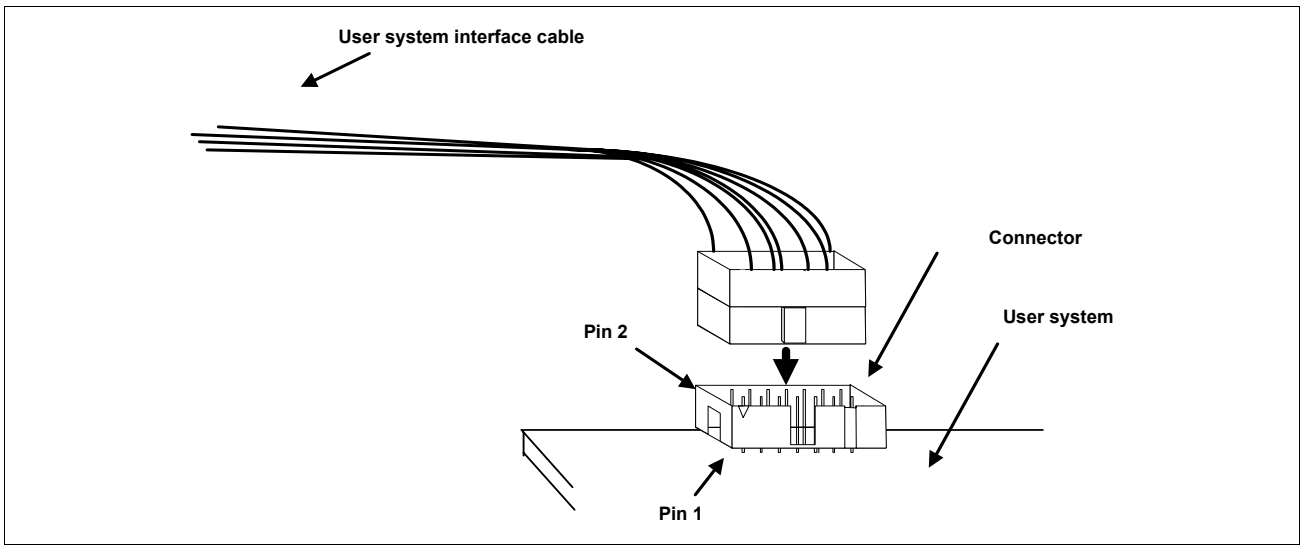
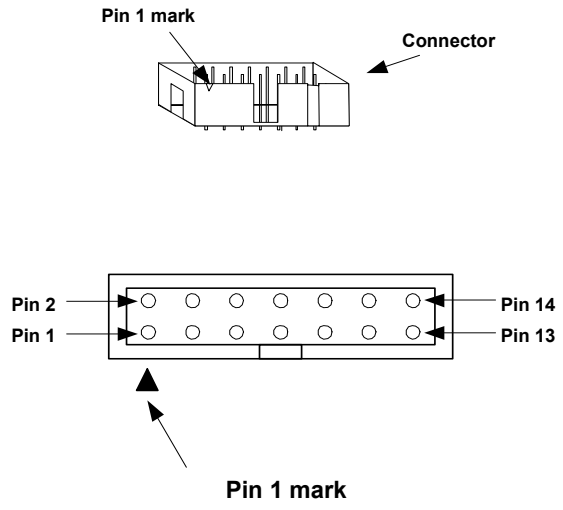


Figure 1.1 Connecting the User System Interface Cable to the User System

Notes: 1. Do not place any components within 3 mm of the connector.
2. When the emulator is used in the writer mode, connect the emulator similarly to the user system.

Section 2 Pin Assignments of the E8 Connector

Figure 2.1 shows the pin assignments of the connector.



Pin NO.	R8C/14, 15, 16, 17 MCU signals
1	N.C.
2	Vss
3	N.C.
4	Vss
5	N.C.
6	Vss
7	**MODE**
8	Vcc
9	N.C.
10	Vss
11	N.C.
12	Vss
13	**$\overline{\text{RESET}}$**
14	Vss

Figure 2.1 Pin Assignments of the E8 Connector

Section 3 Example of E8 Connection

Figure 3.1 shows the connecting example.

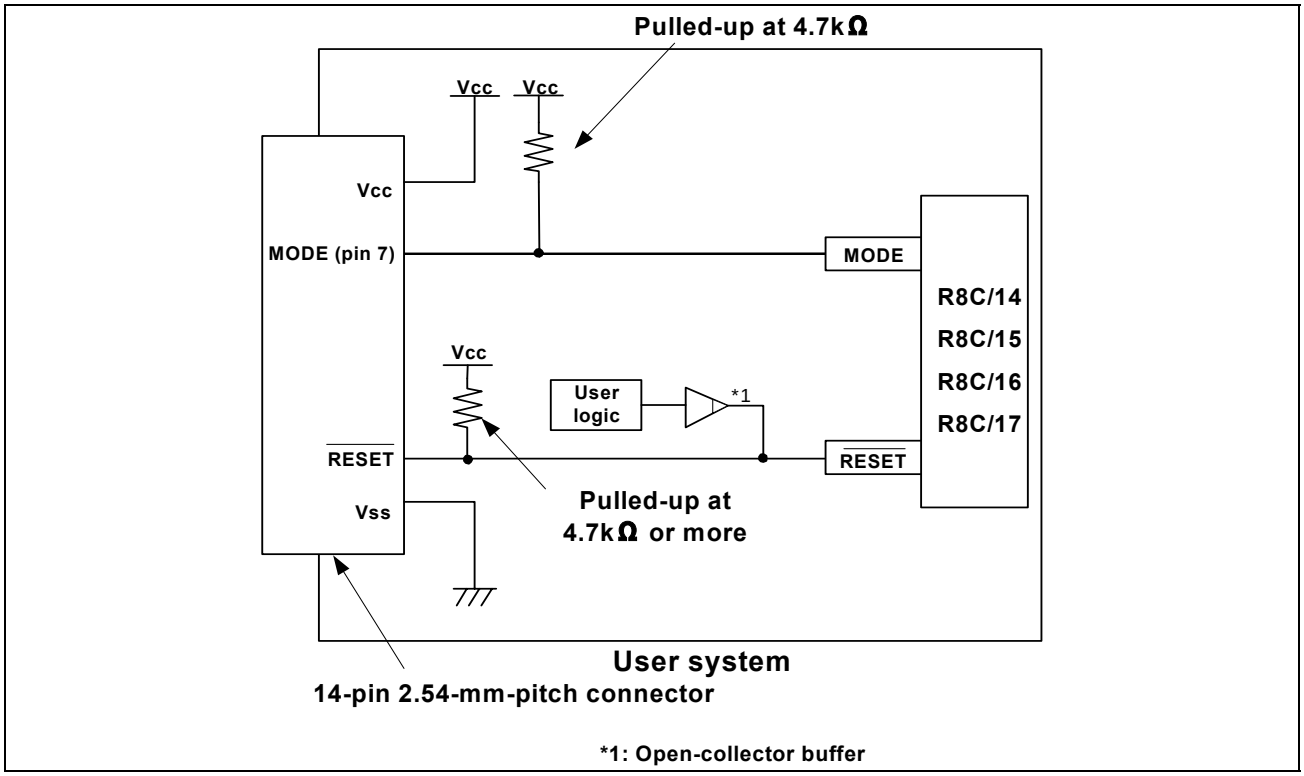


Figure 3.1 Example of E8 Connection

In the ‘Writing Flash memory’ mode, where the user program is simply written to the flash memory, the specification of connection between the E8 and the MCU is the same as that shown in Figure 3.1.

Notes: 1. The E8 emulator uses the MODE pin for the MCU control and the forced break control. Connect the emulator to the MCU pins through pull-up.

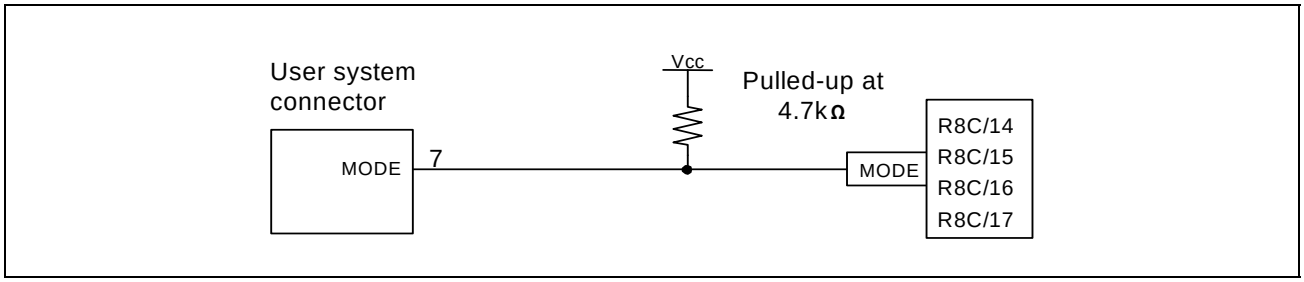


Figure 3.2 Connection of E8 Emulator and MODE Pin

2. The $\overline{\text{RESET}}$ pin is used by the emulator. Create the following circuit by connecting the open-collector output buffer so that reset input can be accepted from the emulator.

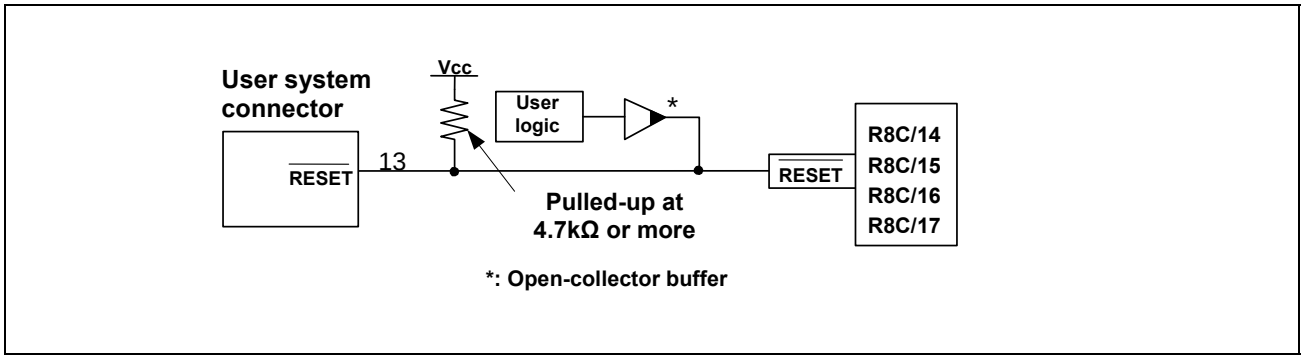


Figure 3.3 Example of a Reset Circuit

3. Connect Vss and Vcc with the Vss and Vcc of the MCU, respectively.
4. Connect nothing with N.C.
5. The amount of voltage permitted to input to Vcc must be within the guaranteed range of the microcomputer.

6. Figure 3.4 shows the interface circuit in the emulator. Use this figure as a reference when determining the pull-up resistance value.

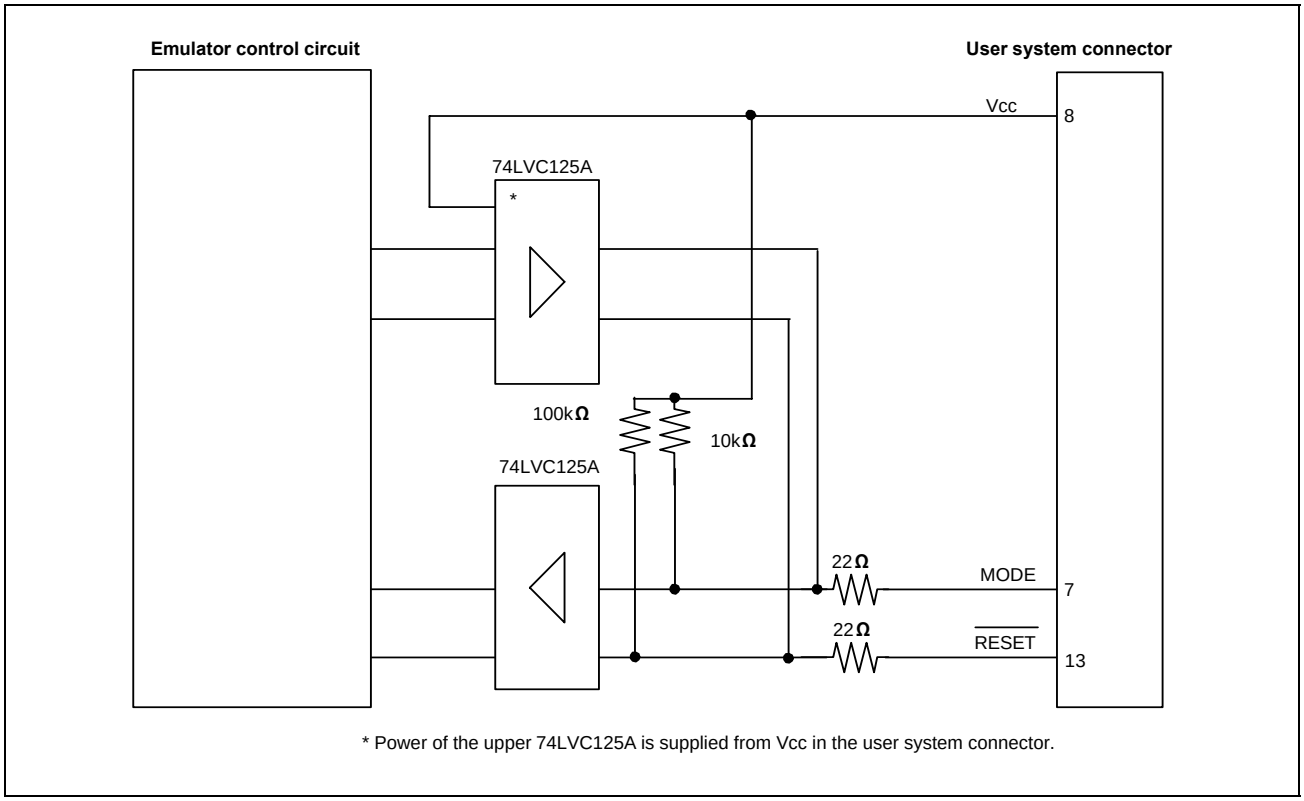


Figure 3.4 Interface Circuit in the Emulator (Reference)

Section 4 Differences between the MCUs and the Emulator

1. Program area for the E8 emulator

   Table 4.1 lists the program area for the E8 emulator.

   Do not change this area, otherwise the E8 emulator will not operate normally. In this case, restart the HEW with the 'Download emulator firmware' mode.

Table 4.1 Program Area for the E8 Emulator

Group	Type Number	ROM Size		Program Area for E8 Emulator	
		Programming Area	Data Area	Vector Area	ROM Area
R8C/14	R5F21142	8 KB	-	FFE4h-FFE8h, FFE8h-FFEBh, FFECh-FFEFh, FFF4h-FFF7h, FFF8h-FFFBh	-
	R5F21143	12 KB	-		-
	R5F21144	16 KB	-		-
R8C/15	R5F21152	8 KB	2 KB		-
	R5F21153	12 KB	2 KB		-
	R5F21154	16 KB	2 KB		2400h-27FFh or [Note] C000h-C7FFh
R8C/16	R5F21162	8 KB	-		-
	R5F21163	12 KB	-		-
	R5F21164	16 KB	-		-
R8C/17	R5F21172	8 KB	2 KB		-
	R5F21173	12 KB	2 KB		-
	R5F21174	16 KB	2 KB		2400h-27FFh or [Note] C000h-C7FFh

Note: If your MCU is R5F21154 or R5F21174, the dialog box shown in Figure 4.1 is displayed when starting up the HEW. Select the location of a program for the E8 emulator with this dialog box.

The location of the program area is 2400h-27FFh and C000h-C7FFh when selecting 'Data Flash Area' and 'User Flash Area', respectively.

When the HEW is started with the 'Does not download emulator firmware' mode, select the area where the firmware has been written to previously.

Figure 4.1 [Firmware Location] Dialog Box

2. ID code of flash memory

When the 7 bytes ID code (Table 4.2) written to the flash memory is other than FFh, FFh, FFh, FFh, FFh, FFh, FFh, input the ID code into the dialog box shown in Figure 4.2 which is displayed when starting up the HEW.

When debugging in 'Download emulator firmware' mode or 'Does not download emulator firmware' mode, FFh, FFh, FFh, FFh, FFh, FFh, FFh is written into the ID code area regardless of the contents of the user program. In 'Writing flash memory' mode, the contents of the user program are input into the ID code area.

Table 4.2 ID Code Storage Area of R8C/14, 15, 16, 17

Address	Description
FFDFh	First byte of ID code
FFF3h	Second byte of ID code
FFEBh	Third byte of ID code
FFEFh	Fourth byte of ID code
FFF3h	Fifth byte of ID code
FFF7h	Sixth byte of ID code
FFFBh	Seventh byte of ID code

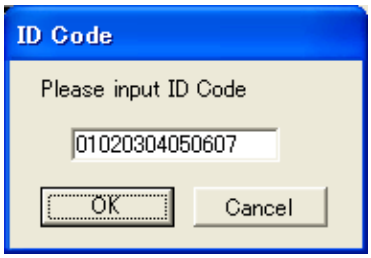


Figure 4.2 [ID Code] Dialog Box

[Note on Writing in Flash memory mode]

When the ID code is specified by the -ID option of the lmc30, download the MOT file or HEX file. When the X30 file is downloaded, the ID code is not effective. When downloading the X30 file, specify the ID code using an assembler directive command such as ".BYTE". The file to which the ID code specified by the assembler directive command ".ID" is output varies depending on the version of the assembler. For details, refer to the user's manual of the assembler.

3. When the emulator system is initiated, it initializes the general registers and part of the control registers as shown in Table 4.3.

Table 4.3 Register Initial Values at Emulator Power-On

Status	Register	Initial Value
Emulator Power-On	PC	Reset vector value in the vector address table
	R0 to R3 (bank 0, 1)	0000h
	A0, A1 (bank 0, 1)	0000h
	FB (bank 0, 1)	0000h
	INTB	0000h
	USP	0000h
	ISP	05FFh
	SB	0000h
	FLG	0000h

4. Operation clock during a break

   During a user program break, the emulator operates changing the CPU clock to the internal high-speed on-chip oscillator (approx. 8 MHz). However, the peripheral features operate with the clock specified by the user program.

5. Reset

   To reset the MCU when debugging by the E8 emulator, select [Debug] -> [Reset CPU] or use the RESET command. If the emulator is reset differently, the E8 cannot be controlled.

6. Memory access during emulation execution

   When referring or modifying the memory contents, the user program is temporarily halted. For this reason, realtime emulation cannot be performed.

7. The emulator communicates with the MCUs by using the MODE and $\overline{\text{RESET}}$ pins.

8. The power consumed by the MCU increases by several mA or over 10 mA. This is because the user power supply drives one 74LVC125A to make the communication signal level match the user-system power-supply voltage.

9. The emulator uses up to four-word stack pointer when a user program breaks. Accordingly, reserve the four-word addresses for the stack area.

10. When debugging, the flash memory is frequently re-written by the E8 emulator. Therefore, do not use an MCU that has been used for debugging.
    Also, as the program for the E8 emulator is written into the MCU while debugging, do not save the contents of the MCU's flash memory that have been used for debugging or use them as the ROM data for products.

11. SFR used by the program for the E8 emulator
    As the SFR listed in Table 4.4 is used by the program for the E8 emulator, they are not initialized by selecting [Debug] -> [Reset CPU] or with the RESET command. If their contents are referred to, a value that has been set in the program for the E8 emulator will be read.
    Do not change the registers by other than the user program, otherwise the E8 cannot be controlled.

Table 4.4 SFR Used by Program for E8 Emulator

Address	Register	Symbol
000Ah	Protect register	PRCR
0020h	High-speed on-chip oscillator A control register 0	HRA0
0021h	High-speed on-chip oscillator A control register 1	HRA1
00B0h	UART transmission control register 2	UCON

12. Reserved area
    The addresses not specified in the Hardware Manual for R8C/14, R8C/15, R8C/16 and R8C/17 Groups are reserved area. Do not change the contents. Otherwise, the E8 emulator cannot be controlled.

13. Debugging in the stop mode or wait mode
    When using the stop mode or wait mode on a user program, firstly disable the automatic update in the watch window or fix the display in the memory window so that the memory access will not occur during execution. In addition, do not operate the window until the program stops at the breakpoint by setting the breakpoint at the processing unit where the stop mode or wait mode is cancelled.

14. Debugging of a watchdog timer
    During the program for the E8 emulator operation, the watchdog timer is being refreshed. Note that if a memory is accessed via the memory reference or modification, the watchdog timer will be refreshed through the intervention of the program for the E8 emulator.

15. Peripheral I/Os during a break
    During a break, although interrupts are not accepted, peripheral I/Os continue to be operated. For example, a timer interrupt is not accepted although counting a timer is continued when a user program is stopped by a break after operating a timer.

16. Exceptional step operation

a) Software-interrupt instruction

STEP operation cannot be performed by continuously executing the internal processing of instructions (undefined, overflow, BRK, and INT) which generates a software interrupt.

<Example>

```
            NOP
            NOP
            INT#3
            NOP          ⤵ Passes through if the STEP operation is carried out.
            JMP MAIN
INT_3:
            NOP  ←—— The address at which the program should be stopped.
            NOP
            NOP
            REIT
```

b) INT instruction

Debugging of the program using the INT instruction should be used with the GO command by setting a software break for the internal processing of the INT instruction.

<Example>

```
            NOP
            INT  #3
            NOP                    ↓ Execution with the GO command
            JMP  MAIN
INT_3:
NOP  Break  ←——
NOP
REIT
```

17. “Run to cursor” function

The "Run to cursor" function is realized by using an address match break. Therefore, when you execute the "Run to cursor" command, all the address match breaks and the hardware breaks you set become invalid, while all the PC breaks remain valid.

18. Note on PC break point

When downloading a user program after changing it, the set address of PC break may not be corrected normally depending on the changes. After downloading a user program, please check the setting of PC break by event point window and reset it.

19. Note on setting the break condition of the hardware breaks

When selecting the normal address bus (selecting [Address] radio button of [Break condition] dialog box) as a condition for the address bus, do not set the following addresses. Otherwise, a malformed break may occur.

- Address in the interrupt vector table
- Address set in the interrupt vector table (interrupt routine start address)
- Branched address of the branching instruction

Both fixed vector table and variable vector table are included with the interrupt vector table above.

20. Note on debugging in CPU rewrite mode

Do not halt the user program after setting the CPU rewrite mode until releasing it. If you do so, the E8 emulator may run out of control. Cancel the automatic renewal in the watch window in advance and select fixing display in the memory window to prevent a memory access from occurring while executing the user program.

To check the data after executing the CPU rewrite mode, halt the program after releasing the CPU rewrite mode and see the memory window etc.

Section 5 Applicable Tool Chain and Partner Tools

With the R8C/Tiny Series E8 emulator, you can debug a module created by the inhouse tool chain and third-party products listed in Table 5.1 below.

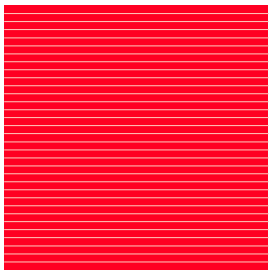
Table 5.1 Applicable Tool Chain and Partner Tools

Tool chain	NC30WA V.5.20 Release 1 or later NC8C V.5.30 Release 1 or later
Partner tools	Tasking CM16 V.2.3 IAR EWM16C V.2.12

E8 Emulator
Additional Document for User's Manual
Notes on Connectiong the R8C/14, R8C/15, R8C/16, and R8C/17

Publication Date:	Jul. 16, 2004	Rev.1.00
	Sep. 16, 2004	Rev.2.00
	Nov. 1, 2004	Rev.3.00
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.	
Edited by:	Microcomputer Tool Development Department Renesas Solutions Corp.	

E8 Emulator
Addtional Document for User's Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan